Research Paper | Computer Science | E-ISSN No : 2454-9916 | Volume : 2 | Issue : 3 | March 2016

DEPLOYING AND RESEARCHING HADOOP ALGORITHMS ON VIRTUAL MACHINES AND ANALYZING LOG FILES

Suyash S. Sathe | Ankit N. Pokharna | Karan R. Hule | Akshay D. Lagad

ABSTRACT

The user behaviors analysis using logs under the big data environment is attractive to the industry profitability for that it can discover the user behaviors to the potential customers. However, the user behaviors are dynamic which is difficult to capture the users' comprehensive behaviors in a single device by capturing or collecting the static dataset. Specially, the increase of the users, network traffic and network services bring many challenges such as fast data collection, processing and storage. Therefore, we propose and implement a log analysis system in this paper, which is based on the hadoop distribution platform to capture the traffic and analyze the user & machine behaviors, in terms of the search keywords, user shopping trends, website posts and replies, and web visited history to acquire the uses' dynamic behaviors. To evaluate our system, we capture the logs in the systems, and the results show that our system can capture the users' long-term behaviors and acquire the user behaviors in detail. in computer log management and intelligence, log analysis (or system and network log analysis) is an art and science seeking to make sense out of computer-generated records (also called log or audit trail records). The process of creating such records is called data logging.

1. INTRODUCTION

This paper is designed to help developers, DevOps engineers, and operations teams that run and manage applications on top of virtual machines to effectively analyze their log data to get visibility into application layers, operating system layers, and different virtual machines services. This booklet is a step-by-step guide to retrieving log data from all cloud layers and then visualizing and correlating these events to give a clear picture of one's entire virtual machines infrastructure.

Cloud applications are inherently more distributed and built out of a series of components that need to operate together to deliver a service to the end user successfully. Analyzing logs becomes imperative in cloud environments because the practice allows relevant teams to see how all of the building blocks of a cloud application are orchestrated independently and in correlation with the rest of the components.

It is the most common log analytics platform in the world. it is used by companies including Netflix, LinkedIn, Facebook, Google, Microsoft, and cisco is an open source stack of three libraries (elastic search, log analysis, and business analytics) that parse, index, and visualise log data (and, yes, it's free).

So instead of going through the challenging task of building a production-ready stack internally, users can sign up and start working in a matter of minutes. In addition, proposed systems as a service includes alerts, multi-user, and role-based access, and unlimited scalability. On top of providing an enterprise-grade platform as a service, proposed system employs unique machine-learning algorithms to automatically surface critical log events before they impact operations, providing users with unprecedented operational visibility into their systems.

➢ SERVER LOG DATA

Large enterprises build, manage and protect their own proprietary, distributed information networks. Server logs are the computer generated records that report data on the operations of those networks. They are like the ekg readings for the network: when there's a problem, it's one of the first places the it team looks for a diagnosis. But no it administrator sits down to read server logs with his morning coffee. The volume is massive, and most often those logs don't matter. However, every once in a while a handful of those logs can be very, very important. The two most common use cases for server log data are network security breaches and network compliance audits. In both of these cases, server log information is vital for both rapid, efficient problem resolution and also longer term forensics and resource planning.

The hadoop ecosystem has made common that :

- Dedicated physical servers.
- Hadoop compute and storage on the same physical machine.
- Hadoop has to be on direct attached storage.

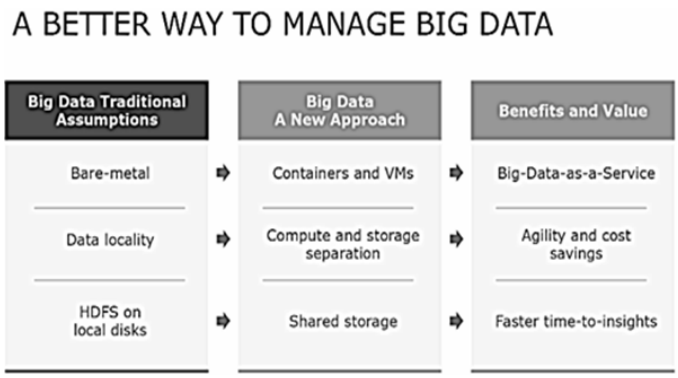

Fig.1.0

In today's distributed architecture, it's important to store all the logs from your application, the OS and the infrastructure provider in one place, In order to properly monitor and be able to quickly troubleshoot issues. Hadoop keeps all the logs in one place and all are synchronized by time allowing you to quickly get a holistic view of your entire environment.

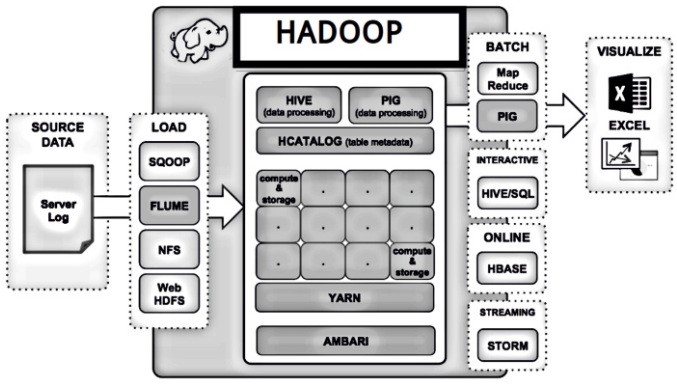

Fig.1.1

➢ Potential Uses of Server Log Data

IT organizations use server log analysis to answer questions about:

Security – For example, if we suspect a security breach, how can we use server log data to identify and repair the vulnerability?

Compliance – Large organizations are bound by regulations such as HIPAA and Sarbanes-Oxley. How can IT administrators prepare for system audits?

In this Paper, we will focus on a network security use case. Specifically, we will look at how Apache Hadoop can help the administrator of a large enterprise network diagnose and respond to a distributed denial-of-service attack.

Data analysis is only half the battle; getting the data into a Hadoop cluster is the first step in any Big Data deployment. Apache Flume uses an elegant design to make data loading easy and efficient.

In this Paper, we will focus on a network security use case. Specifically, we will look at how Apache Hadoop can help the administrator of a large enterprise network diagnose and respond to a distributed denial-of-service attack.

Data analysis is only half the battle; getting the data into a Hadoop cluster is the first step in any Big Data deployment. Apache Flume uses an elegant design to make data loading easy and efficient.

hdfs-agent.sources= netcat-collect

hdfs-agent.sinks = hdfs-write

hdfs-agent.channels= memory-channel

hdfs-agent.sources.netcat-collect.type = netcat

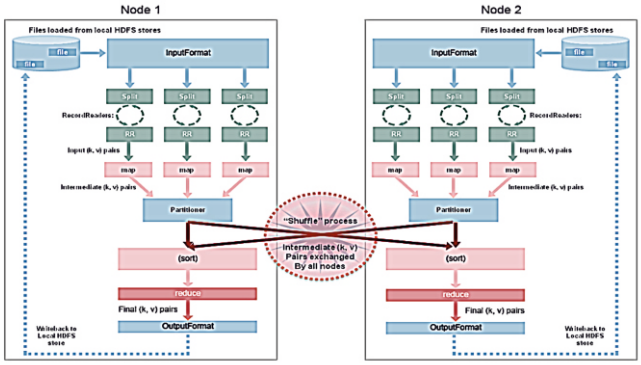

fig.1.2

hdfs-agent.sources.netcat-collect.bind = 127.0.0.1

hdfs-agent.sources.netcat-collect.port = 11111
hdfs-agent.sinks.hdfs-write.type = hdfs

hdfs-agent.sinks.hdfs-write.hdfs.path =
hdfs://namenode_address:8020/path/to/flume_test

hdfs-agent.sinks.hdfs-write.rollInterval = 30

hdfs-agent.sinks.hdfs-write.hdfs.writeFormat=Text

hdfs-agent.sinks.hdfs-write.hdfs.fileType=DataStream

hdfs-agent.channels.memoryChannel.type = memory

fig.1.1

hdfs-agent.channels.memoryChannel.capacity=10000

hdfs-agent.sources.netcat-collect.channels=memoryChannel

hdfs-agent.sinks.hdfs-write.channel=memoryChannel

fluming agent -f /path/to/sample_agent.conf -n hdfs-agent

#sender configuration

avro-agent.sinks= avro-sink

avro-agent.sinks.avro-sink.type=avro

avro-agent.sinks.avro-sink.host=remote.host.com

avro-agent.sinks.avro-sink.port=11111

#receiver configuration on remote.host.com

hdfs-agent.sources=avro-source

hdfs-agent.sources.avro-source.type=avro

hdfs-agent.sources.avro-source.bind=0.0.0.0

hdfs-agent.sources.avro-source.port=11111

hdfs-agent.sources.avro-source.channels=memoryChannel

hdfs-agent.channels=mchannel1 mchannel2

hdfs-agent.sources.netcat-collect.selector.type = replicating

hdfs-agent.sources.r1.channels = mchannel1 mchannel2

hdfs-agent.sources.netcat-collect.interceptors = filt_int

hdfs-agent.sources.netcat-collect.interceptors.filt_int.type=regex_filter

hdfs-agent.sources.netcat-collect.interceptors.filt_int.regex=^echo.*

hdfs-agent.sources.netcat-collect.interceptors.filt_int.excludeEvents=true

Log data is a definitive record of what's happening in every business, organization or agency and it's often an untapped resource when it comes to troubleshooting and supporting broader business objectives.

Proposed system provides the industry-leading software to consolidate and index any log and machine data, including structured, unstructured and complex multi-line application logs. You can collect, store, index, search, correlate, visualize, analyze and report on any machine-generated data to identify and resolve operational and security issues in a faster, repeatable and more affordable way. It's an enterprise ready, fully integrated solution for log management data collection, storage and visualization.

Ad hoc queries and reporting across historical data can also be accomplished without third-party reporting software. Proposed system software supports log data enrichment by providing flexible access to relational databases, field delimited data in comma-separated value (.CSV) files or to other enterprise data stores such as Hadoop or NoSQL. Proposed system software supports a wide range of log management use cases including log consolidation and retention, security, IT operations troubleshooting, application troubleshooting and compliance reporting..

- Index, search and correlate any data for complete insight across your infrastructure

- Drill down and up and pivot across data to quickly find the needle in the haystack

- Turn searches into real-time alerts, reports or dashboards with a few mouse clicks

2. TECHNIQUES FOR IMPLEMENTATION

Step 1. Configure and start apache flume.

- First, login in to the Sandbox using the Ambari user interface which can be found at http://sandbox.hortonworks.com:8080.

- Note that if the above doesn't work you can either add sandbox.hortonworks.com to your /etc/hosts file or you can trying using http://localhost:8080 or http://127.0.0.1:8080.

- Once you've logged in you'll need to use the Ambari views dropdown menu and select Local Files. This a view of the Sandbox VM's file system (not HDFS).

Step 2. Start flume service

- Click on flume servise and click start (If not started)

Step 3. Generate the server log data

- Now that Flume is running, we will use a Python script to generate the server log data, and then create an Catalog table from the data.

Step 4. Import the server log data into Excel.

- In Windows, open a new Excel workbook, then select Data > From Other Sources > From Microsoft Query.

- On the Choose Data Source pop-up, select the Hortonworks ODBC data source you installed previously, then click OK.

- After the connection to the Sandbox is established, the Query Wizard appears. Select the "firewall_logs" table in the Available tables and columns box, then click the right arrow button to add the entire "firewall_logs" table to the query. Click Next to continue.

- On the Filter Data screen, click next to continue without filtering the data.

- On the Sort Order screen, click next to continue without setting a sort order.

- On the Import Data dialog box, click OK to accept the default settings and

import the data as a table.

- The imported query data appears in the Excel workbook.
- The Hortonworks ODBC driver enables you to access Hortonworks data with Excel and other Business Intelligence (BI) applications that support ODBC.

Step 5. Visualize the sentiment data using Excel power point.
Ultimately we will get the window on Excel sheet for analyzing server log data.

3. FUTURE WORK

Till now whatever work has been done for visualization of different logs data is on sample of data. To integrate this data together in structure format and visualize this as a one quantity is important. So instead of visualizing as sampled logs data, integrated logs data visualization is of prime importance.

4. ADVANCEMENT

Data coming to the organization is from different sources like click stream, networking site, log files , web logs, sensors, emails which are categorized into structured, semi structured and unstructured.

Server logs, web logs and other logs data are pushed into the archival. But nowadays companies planned strategy to fetch logs data from archival and make use of it. So visualization of logs data is came into role. We can do processing on any of logs data for visualization and then Business Intelligence team can make use of this information for betterment of the organization.

5. CONCLUSION

There are lots of ways to acquire Big Data with which to fill up a Hadoop cluster, but many of those data sources arrive as fast-moving streams of data. Fortunately, the Hadoop ecosystem contains a component specifically designed for transporting and writing these streams: Apache Flume. Flume provides a robust, self-contained application which ensures reliable transportation of streaming data. Flume agents are easy to configure, requiring only a property file and an agent name. Moreover, Flume's simple source-channel-sink design allows us to build complicated flows using only a set of Flume agent's process of acquiring Big Data for our Hadoop clusters, doing so as easy and fun as taking a log ride.

6. REFERANCES

1. Mahout: Scalable machine-learning and data-mining library.
   http://mapout.apache.org, 2010.
2. http://hadoop.apache.org/
3. http://pig.apache.org/